Incremental training of Markov mixture models

Andreas Kakoliris and Konstantinos Blekas

Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece
E-mail: {akakolir, kblekas}@cs.uoi.gr

Abstract. This paper presents an incremental approach for training a Markov mixture model to a set of sequences of discrete states. Starting from a single Markov model that captures the background information, at each step a new component is added to the mixture in order to improve the data fit. This is done by making at first an exploration of a relevant parametric space to initialize the new component, based on an extension of the k-means algorithm. Then, by performing a two-stage scheme of the EM algorithm, the new component is optimally incorporated to the body of the current mixture. To assess the effectiveness of the proposed method, we have conducted experiments with several data sets and we make a performance comparison with the classical mixture model.

1 Introduction

Sequential data analysis is an important research area with a wide range of applications, such as web log mining, bioinformatics, speech recognition, robotics, natural language processing and many others. Since clustering can be seen as a fundamental tool in understanding and exploring a data set, several attempts have been made on the task of clustering sequential data of discrete states [1–4]. In model-based clustering approaches a flexible and powerful scheme used is through mixture models [5]. It is assumed that each cluster is described by a generative model and the aim of clustering is to find an optimal set of such models in order to best fit the data. Markov models [6] provide an efficient method for modeling sequential data. In most of the these approaches, the EM algorithm [7] is used for estimating the parameters of the Markov mixture models. Since the EM algorithm has the drawback to be dependent on the initial values of the mixture parameters, several methods have been introduced to reduce this effect. In [3] for example, a noisy-marginal scheme is proposed by perturbing the parameters of a single model to obtain K copies of it. An alternative approach is presented in [1], where an agglomerative clustering technique is applied together with a suitable distance function for sequences, in order to initialize the K parametric models of the mixture. Many efforts have been made recently to address visualization capabilities of clustering approaches using Markov models [3, 8–10]. In this spirit, the behavior of the sequences within clusters can be displayed and an explanatory analysis for the dynamics of data can be provided.

In this paper we propose an incremental approach for training Markov mixture models. Borrowing strength from recent advances on mixture models [11,

12], our method performs a systematic exploration of the parameter space and simultaneously tries to eliminate the dependence of the EM algorithm on the initialization. The method starts with a single Markov model that fits all sequences, and sequentially adds new components to the mixture following three major steps. At first we initialize the new inserted Markov model by searching over a parametric likelihood space. The latter is specified by a set of candidate models that have been constructed through the use of an adaptation of the classical k-means algorithm for treating sequential data. This is the initialization step. Then, we perform a partial EM scheme allowing the adjustment of only the new model parameters. Finally, the new component is optimally incorporated to the current mixture by normally applying the EM algorithm and thus best fitting the new mixture model with the data. The procedure stops when reaching a number of K components. We have tested our training method on a suite of artificial and real benchmarks taking into account a variety of cases with excellent results. During the experiments we have evaluated the proposed scheme in terms of its capability to fit the data and measure its robustness. Comparative results have been also obtained with the classical Markov mixture model under two schemes for initialization.

In section 2 we give the basic scheme of the Markov mixture models, while section 3 describes the proposed approach for incremental training. In section 4 we present experimental results and finally, in section 5 we give some concluded remarks.

2 Markov mixture models

Consider a dataset $X = \{X_1, \ldots, X_N\}$, where each data point $X_i = (X_{il})_{l=1}^{L_i}$ is a sequence of length L_i observed states. We further assume that each state takes values from a discrete alphabet of M symbols, i.e. $X_{il} \in \{1, \ldots, M\}$. The clustering problem is to find K disjoint subsets of X, called clusters, containing sequences with common properties. In this study we consider that every cluster corresponds to a generative model that fits well the observed data that supports.

Mixture models represent an efficient architecture that is particularly suitable for clustering. It assumes that data have been generated from a mixture model with K components according to the following density function

$$f(X_i|\Theta_K) = \sum_{j=1}^{K} \pi_j p(X_i|\theta^j) \ , \tag{1}$$

where $\Theta_K = \{\pi_j, \theta^j\}$ denotes the set of the mixture parameters. In particular, the parameters $\pi_j = P(j)$ determine the prior probabilities of the K components satisfying that $\sum_{j=1}^{K} \pi_j = 1$. Moreover, every component has a probability distribution function $p(X_i|\theta^j)$, whose parameters θ^j are unknown. A natural way for modeling sequential data is through the first-order Markov model, defined by the initial states probabilities $\theta^j_{0m} = P(X_{i1} = m)$, as well as the transition probabilities $\theta^j_{nm} = P(X_{i,l+1} = m|X_{il} = n)$ from a state n to another state m,

$n, m = 1, \ldots, M$. Thus, each model parameter θ^j is a stochastic matrix with a set of $M+1$ rows (multinomial distributions), holding that $\sum_{m=1}^{M} \theta_{nm}^j = 1$, $\forall n = 0, \ldots, M$. The density function for the jth component is then written as

$$p(X_i|\theta^j) = \theta_{0,X_{i1}}^j \prod_{l=1}^{L_i-1} \theta_{X_{il},X_{i,l+1}}^j = \prod_{m=1}^{M} (\theta_{0m}^j)^{\gamma_i(m)} \prod_{n=1}^{M} \prod_{m=1}^{M} (\theta_{nm}^j)^{\delta_i(n,m)}, \quad (2)$$

where $\gamma_i(m) = \begin{cases} 1 \text{ if } X_{i1} = m \\ 0 \text{ otherwise} \end{cases}$ and $\delta_i(n,m)$ defines the number of transitions from state n to state m in the sequence X_i. Following the Bayes rule, we can then associate every sequence X_i to the cluster j that has the maximum posterior probability value $P(j|X_i) = \frac{\pi_j p(X_i|\theta^j)}{f(X_i|\Theta_K)}$. The clustering problem is then equivalent to estimating the mixture model parameters Θ_K, by maximizing the log-likelihood function arisen from the model. Furthermore, we can introduce non-informative Dirichlet priors of the form $p(\theta_n^j|a_n^j) = \frac{\Gamma(\sum_{m=1}^{M}(a_{nm}^j+1))}{\prod_{m=1}^{M}\Gamma(a_{nm}^j+1)} \prod_{m=1}^{M}(\theta_{nm}^j)^{a_{nm}^j}$, where the parameter a_n^j is a M-vector with components $a_{nm}^j > 0$. The derived *maximum a-posteriori* (MAP) log-likelihood function is then given by

$$L(X|\Theta_K) = \sum_{i=1}^{N} \log f(X_i|\Theta_K) + \sum_{j=1}^{K} \sum_{n=0}^{M} \log p(\theta_n^j|a_n^j) \ . \quad (3)$$

It must be noted that the Dirichlet parameters a_n^j were common to every component j and set equal to a small proportion (e.g. 10%) of the corresponding maximum likelihood (ML) estimated multinomial parameter values of the single Markov model that fits the data set X (using relative frequencies of states). The latter from now on it will be referred to as "single ML-estimated Markov model".

The EM algorithm [7] is an efficient framework for estimating the mixture model parameters. It requires the computation of the conditional expectation values z_{ij} (posterior probabilities) of the hidden variables during the E-step $z_{ij}^{(t)} = \frac{\pi_j^{(t)} p(X_i|\theta^{j\,(t)})}{\sum_{j'=1}^{K} \pi_{j'}^{(t)} p(X_i|\theta^{j'\,(t)})}$, while at the M-step the maximization of the log-likelihood function of the complete dataset is performed. This leads to the following updated equations for the mixture model parameters:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)}}{N} \ , \ \theta_{nm}^{j\ (t+1)} = \begin{cases} \dfrac{\sum_{i=1}^{N} z_{ij}^{(t)} \gamma_i(m) + a_{0m}^j}{\sum_{i=1}^{N} z_{ij}^{(t)} + \sum_{m'=0}^{M} a_{0m'}^j} & \text{, if } n = 0 \\ \dfrac{\sum_{i=1}^{N} z_{ij}^{(t)} \delta_i(n,m) + a_{nm}^j}{\sum_{i=1}^{N} z_{ij}^{(t)} \sum_{m'=1}^{M} \delta_i(n,m') + \sum_{m'=1}^{M} a_{nm'}^j} & \text{, if } n > 0 \end{cases} \quad (4)$$

The EM algorithm guarantees the convergence of the log-likelihood function to a local maximum satisfying all the constraints of the parameters. However, the great dependence on the initial parameter values may drastically effect its performance [5]. In the next section we present an incremental approach for building a Markov mixture models that eliminates this problem of poor initialization.

3 Incremental mixture training

The proposed method starts with a simple model with one component that comes from the single ML-estimated Markov model of the whole dataset X. At each step a new component is added to the mixture by performing a combined scheme of searching for good initial estimators and for fine local tuning its parameters. It must be noted that a same in nature strategy have been also presented in [11] and [12] for Gaussian mixture models and for discovering patterns in biological sequences, correspondingly.

Lets assume that we have already constructed a k-length mixture model with Θ_k parameters. By inserting a new component, the resulting mixture can take the following form

$$f(X_i|\Theta_k, \pi^*, \theta^*) = (1 - \pi^*) f(X_i|\Theta_k) + \pi^* p(X_i|\theta^*) \ . \tag{5}$$

where $\pi^* \in (0, 1)$ is the prior probability of the new component. The above scheme can be viewed as a two-component mixture model, where the first one captures the current mixture with density function $f(X_i|\Theta_k)$ and the second one the new Markov model that has a density function $p(X_i|\theta^*)$ with an unknown stochastic matrix θ^*.

If we fix the parameters of the old mixture model Θ_k, we can then maximize the resulting log-likelihood function $\mathcal{L}_k$ of the above two-components mixture with respect only to the new model parameters $\{\pi^*, \theta^*\}$:

$$\mathcal{L}_k = \sum_{i=1}^{N} \log\{(1 - \pi^*) f(X_i|\Theta_k) + \pi^* p(X_i|\theta^*)\} + \sum_{n=0}^{M} \log p(\theta_n^*|a_n) \ . \tag{6}$$

In this light, we can apply the EM algorithm for estimating only the parameters of the new model, namely as *partial EM*. This results into obtaining the following update equations: a) at the E-step

$$\zeta_i^{(t)} = \frac{\pi^{*(t)} p(X_i|\theta^{*(t)})}{(1 - \pi^{*(t)}) f(X_i|\Theta_k) + \pi^{*(t)} p(X_i|\theta^{*(t)})} \ , \tag{7}$$

and b) at the M-step

$$\pi^{*(t+1)} = \frac{\sum_{i=1}^{N} \zeta_i^{(t)}}{N},\ \theta_{nm}^{*\ (t+1)} = \begin{cases} \dfrac{\sum_{i=1}^{N} \zeta_i^{(t)} \gamma_i(m) + a_{0m}}{\sum_{i=1}^{N} \zeta_i^{(t)} + \sum_{m'=0}^{M} a_{0m'}} & , \text{if } n = 0 \\ \dfrac{\sum_{i=1}^{N} \zeta_i^{(t)} \delta_i(n,m) + a_{nm}}{\sum_{i=1}^{N} \zeta_i^{(t)} \sum_{m'=1}^{M} \delta_i(n,m') + \sum_{m'=1}^{M} a_{nm'}} & , \text{if } n > 0 \end{cases} \tag{8}$$

The above partial EM steps offer more flexibility to the general scheme and simplifies the estimation problem during the insertion of a new Markov model to the mixture.

At a second stage, the new component can be incorporated to the body of the current mixture and construct a new mixture $f(X_i|\Theta_{k+1})$ with $k+1$ components. Again, the EM algorithm can be used to maximize the log-likelihood function $L(X|\Theta_{k+1})$ in the new parameter space Θ_{k+1}, following Eqs. 4. The mixture parameters are initialized from the solution of the partial EM, i.e. $\pi_{k+1}^{(0)} = \pi^*$, $\pi_j^{(0)} = (1 - \pi^*)\pi_j$, $\forall j = 1, \ldots, k$, and $\theta_{k+1}^{(0)} = \theta^*$. This iterative procedure is repeated until the desired order K of the Markov mixture model is reached.

3.1 Initializing new model parameters

From the above analysis, a problem that arises is how to initialize properly the new component parameters during the partial EM scheme. This can be accomplished by establishing a parametric search space through a set of K_m candidate Markov models $\{\phi_j\}_{j=1}^{K_m}$. In particular, we perform one step of the partial EM, after initializing the multinomial parameters of the new model with a candidate Markov model ($\theta^{*(0)} = \phi_j$) and the prior probability π^* with the typical value $\pi^{*(0)} = \frac{1}{k+1}$. Finally, we select the solution that corresponds to the maximum value of the log-likelihood function $\mathcal{L}_k$ (Eq. 6) for initializing the parameters $\{\pi^*, \theta^*\}$.

In our study we have used an extension of the known k-means algorithm to create such a set of candidate models. In the general case, the k-means algorithm aims at finding a partition of K_m disjoint clusters C_j to a set of N objects, so as the overall sum of distances between cluster centers μ_j and objects X_i is minimized. In order to adopt this framework in the case of sequential data we need to make some modifications. At first, a distance function between two sequences X_i and X_k must be provided so as to encapsulate an appropriate measure of dissimilarity between data. For this purpose we have used a symmetrized log-likelihood

distance defined as [1]

$$D(i,k) = \frac{1}{2}\{\log p(X_i|\vartheta_k) + \log p(X_k|\vartheta_i)\} \ , \tag{9}$$

where the parameters ϑ_i denote the single ML-estimated Markov model specified by each sequence X_i. Furthermore, at each step t of the k-means algorithm we re-estimate the new center $\mu_j^{(t+1)}$ of every cluster C_j by finding the *medoid* sequence among the sequences that currently supports, i.e. $\mu_j^{(t+1)} = \arg\min_{X_i \in C_j^{(t)}} \sum_{X_k \in C_j^{(t)}} D(i,k)$. At the end of the algorithm, we correspond a Markov model ϕ_j to every cluster C_j, by finding the single (ML-estimated) Markov model that best fits all sequences associated with this cluster. The above scheme creates a pool of K_m candidate models capable for initializing the parameter θ^* during the partial EM steps. As experimental study has shown, the proposed method is not sensitive to the value of K_m. A small proportion of the population size of sequences N (e.g. 5%) is enough for constructing a rich search space with good initial estimators. Another advantage of the proposed k-means algorithm is that is computationally faster than other distance-based clustering schemes (e.g. hierarchical clustering) that can be alternatively applied using the same distance function (Eq. 9).

3.2 The proposed algorithm

The proposed incremental approach for training a mixture of K Markov models can be summarized in the following algorithmic form.

- Set $\Theta_1 = \{\theta^1, \pi_1 = 1\}$ using the single ML-estimated Markov model from the data set X. Use k-means to provide K_m candidate Markov models ϕ_j.
- for $k = 1 : K-1$
  1. $\forall j = 1, \ldots, K_m$ perform one partial EM step (Eqs.7-8) by setting ${\pi^*}^{(0)} = \frac{1}{k+1}$ and ${\theta^*}^{(0)} = \phi_j$. Select the solution that has the maximum log-likelihood value $\mathcal{L}_k$ (Eq. 6).
  2. Perform partial EM (Eqs.7-8) until convergence and estimate new model parameters $\{\pi^*, \theta^*\}$.
  3. Set $\Theta_{k+1} = \Theta_k \cup \{\pi_{k+1}, \theta_{k+1}\}$, where ${\pi_{k+1}}^{(0)} = \pi^*$, ${\pi_j}^{(0)} = (1-\pi^*)\pi_j$ $\forall j \leq k$, ${\theta^{k+1}}^{(0)} = \theta^*$.
  4. Perform general EM (Eqs.4) to maximize $L(X|\Theta_{k+1})$.

4 Experimental results

Several experiments have been made in an attempt to evaluate the performance of the proposed incremental training approach, namely as IMM. Comparative results have been also obtained using two methods for initializing classical Markov mixture models: a) the RMM, that follows the initialization scheme presented in [3] which creates K noisy copies from the single ML-estimated Markov model,

and b) the KMM, that first applies the k-means algorithm as described previously for discovering K clusters ($K_m = K$), and then initializes every component with the single ML-estimated Markov model of every cluster found. In any case, the prior parameters are initially set as $\pi_j = 1/K$. Since both last methods depends on the initialization, twenty (20) runs of the EM algorithm were performed for each data set. We kept records of the mean value and the standard deviation of the log-likelihood. Also, the proposed IMM model was executed only once for fitting a K-order Markov mixture model to each data set.

Table 1. Percentage of times the correct model was detected by the three methods IMM, KMM and RMM.

# symbols (M)	*mixture model*	# components (K)			
		5	8	10	15
5	*IMM*	100 %	100 %	100 %	100 %
	RMM	80.5 %	67.5 %	50 %	25.5 %
	KMM	56 %	49.5 %	31.5 %	7 %
8	*IMM*	100 %	100 %	100 %	100 %
	RMM	67 %	47.5 %	35.5 %	11.5 %
	KMM	50 %	32 %	19 %	7 %
10	*IMM*	100 %	100 %	100 %	100 %
	RMM	74.5 %	45.5 %	28.5 %	10 %
	KMM	47 %	28.5 %	15.5 %	2.5 %
12	*IMM*	100 %	100 %	100 %	100 %
	RMM	70 %	42 %	26.5 %	9.5 %
	KMM	35 %	25 %	11 %	2.5 %
15	*IMM*	100 %	100 %	100 %	100 %
	RMM	75 %	35.5 %	21 %	6 %
	KMM	52 %	20.5 %	9.5 %	1 %

The first series of experiments was carried out using artificial data to evaluate the robustness of our method. We created sets of artificial sequences by sampling from several K-order Markov mixture models using various values for the alphabet size M. In particular, using five and four different values for the parameters K and M, correspondingly, we created ten (10) different datasets for each pair (M, K). In each dataset $N = 1000$ number of sequences were generated of length between 50 and 100 states ($L_i \in [50, 100]$). Since we were aware of the true model that best fit the experimental datasets, we evaluated each method by calculating the percentage of times that the global maximum log-likelihood value was found. Table 1 summarizes the depicted results. The weakness of both the RMM and KMM approaches in obtaining the global maximum value, is obvious, especially in higher values of K. On the other hand, the proposed IMM approach was able to estimate correctly the true model in all cases.

Another series of experiments with artificial sequences has been made using sets of K randomly selected patterns of equal length 50 from an alphabet of

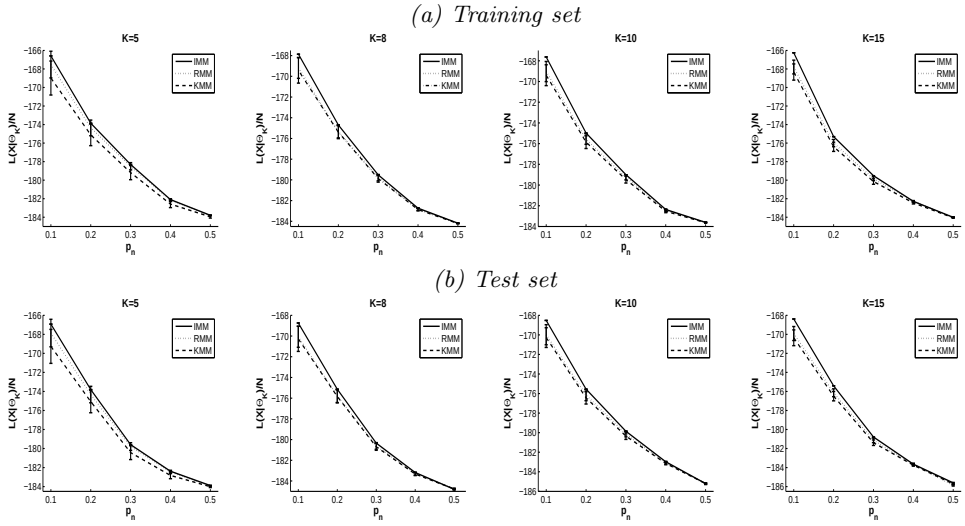

Fig. 1. The log-likelihood values found by the three comparative methods as a function of noise parameter p_n.

M symbols. The data generation mechanism was the following: A pattern is randomly selected at first, and then a noisy copy of it is located at a random position in the sequence. Pattern noise is governed by using a probability p_n for mutation, common to every pattern site. The rest non-pattern sites are filled uniformly from the same alphabet. Using this scheme, two sets of $N = 1000$ sequences of length $L_i \in [50, 100]$ were created; one used for training and another one for testing. As it is clear, this clustering problem is more difficult since the Markov property exists only locally in the sequences and under different levels of noise. Likewise, for each randomly selected pattern family we generated ten (10) different datasets and we evaluated each method in terms of the log-likelihood value found in both training and test sets. Figure 1 illustrates the results obtained with four values of $K = \{5, 8, 10, 15\}$ and five different levels of noise $p_n = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ in the case of $M = 10$ alphabet size. In each diagram, the error bars indicate the standard deviation of the log-likelihood difference between the true model that is known and the model under consideration. Our method was able to achieve a high degree of noise tolerance, since always managed to discover the correct model, even for extremely noisy datasets.

Additional experiments have been performed using the msnbc.com web navigation dataset [3], which is a collection of sequences that corresponds to $M = 17$ page-category views (symbols) of users during twenty-four hour period. Here we have considered only a subset of the total collection containing 4600 sequences of length $L_i \in [40, 100]$. We randomly divided it into two subsets (training / test) of approximately equal size. Figure 2 shows the calculated log-likelihood value per sequence ($L(X|\Theta_K)/N$) as a function of the mixture order K on both

sets. Our method was executed only once until reaching $K = 16$ components. In the case of the RMM and KMM methods we plot the mean value and the standard deviations (error bars) of the log-likelihood over 20 different runs (initializations) of the EM algorithm per each value of K. The proposed method showed an improvement performance with better generalization capabilities on the test set in comparison with the other two approaches. Note that we have repeated this study with different divisions into training and test subsets of this dataset and the results were similar. Finally, in Figure 3 we give an example of the visualization capabilities of clustering sequential data that can be used for identifying user behavior patterns in applications such as web log mining [3, 8]. Each one of the ten images corresponds to a cluster found when applying our method for training a mixture model with $K = 10$ Markov components. By associating every symbol with a unique color, sequences that belong to the same cluster are represented as raws of colored squares.

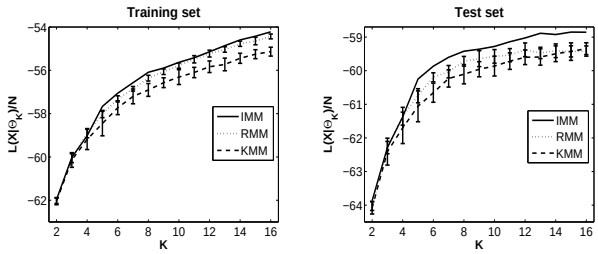

Fig. 2. Application of the three methods to the msnbc.com dataset. The log-likelihood values are calculated for several values of K in the training and the test set.

5 Conclusions

In this paper we have presented an incremental strategy for training Markov mixture models by maximum likelihood on a set of sequences of discrete states. The approach sequentially adds components to a mixture model by performing a combined scheme of the EM algorithm. In order to initialize properly each new component, an efficient parameter search space of Markov models has been constructed. Experiments on a variety of benchmarks have shown the ability of our method to improve the data fit and also demonstrated its generalization capability. The determination of the proper value of K for terminating the incremental procedure can be seen as one of our future studies on this area. Finally, we plan to focus our attention on mixtures of hidden Markov models, since they can be seen as more general probabilistic models for sequential data.

Acknowledgments. This research is partially supported by the EPEAEK II Program.

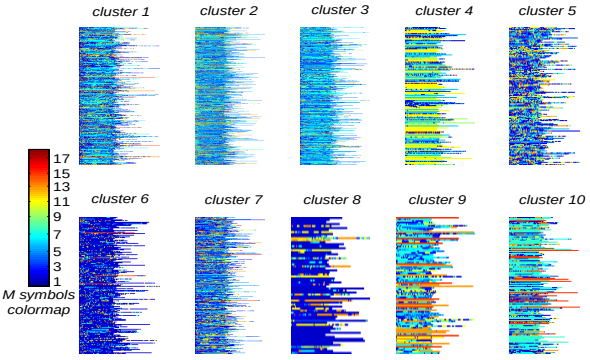

Fig. 3. Visualization of the clustering results ($K = 10$) on the msnbc.com data. Each image represents a cluster of user sessions in a colored raw form.

References

1. P Smyth. Clustering sequences with hidden Markov models. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 648–654. The MIT Press, 1997.
2. G. Ridgeway. Finite discrete markov process clustering. Technical Report MSR-TR-97-24. Microsoft Research, Redmod, WA, 1997.
3. I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, 7(4):399–424, 2003.
4. M. Bicego, V. Murino, and M. Figueiredo. Similarity-based classification of sequences using hidden Markov models. *Pattern Recognition*, 37:2281–2291, 2004.
5. G.M. McLachlan and D. Peel. *Finite mixture models*. New York: John Wiley & Sons, Inc., 2001.
6. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.
7. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39:1–38, 1977.
8. E. Manavoglu, D. Pavlov, and C.L. Giles. Probabilistic user behavior models. In *IEEE International Conference on Data Mining (ICDM'03)*, pages 203–210, 2003.
9. A. Ypma and T.M. Heskes. Automatic categorization of web pages and user clustering with mixtures of hidden Markov models. In *WEBKDD 2002 - Mining web data for discovering usage patterns and profiles*, pages 35–49, Berlin, 2003.
10. P. Tino, A. Kaban, and Y. Sun. A generative probabilistic approach to visualizing sets of symbolic sequences. In *ACM SIGKDD - International Conference on Knowledge Discovery and Data Mining - KDD-2004*, pages 701–706, 2004.
11. N. Vlassis and A. Likas. A greedy EM algorithm for Gaussian mixture learning. *Neural Processing Letters*, 15(1):77–87, 2002.
12. K. Blekas, D.I. Fotiadis, and A. Likas. Greedy mixture learning for multiple motif discovering in biological sequences. *Bioinformatics*, 19(5):607–617, 2003.